NCR-44-006-089

Dr J. W. Lewallen
EP-13
MSC

AERO-ASTRONAUTICS REPORT NO. 79

Get DRA

MODIFIED QUASILINEARIZATION METHOD FOR SOLVING NONLINEAR, TWO-POINT BOUNDARY-VALUE PROBLEMS

by

A. MIELE AND R.R. IYER

RICE UNIVERSITY

1970

Modified Quasilinearization Method for Solving Nonlinear, Two-Point Boundary-Value Problems[1]

by

A. MIELE[2] AND R. R. IYER[3]

Abstract. This paper presents a general method for solving nonlinear, differential equations of the form $\dot{x} - \varphi(x, t) = 0$, $0 \le t \le 1$, subject to boundary conditions of the form $f[x(0)] = 0$, $g[x(1)] = 0$, $h[x(0), x(1)] = 0$. Here, t is a scalar, x and φ are n-vectors, and f, g, h are p, q, r-vectors, with $p + q + r = n$. The method is based on the consideration of the performance index P, the cumulative error in the differential equations and the boundary conditions.

A modified quasilinearization algorithm is generated by requiring the first variation of the performance index δP to be negative. This algorithm differs from the ordinary quasilinearization algorithm because of the inclusion of the scaling factor or stepsize α in the system of variations. The main property of the modified quasilinearization algorithm is the descent property: if the stepsize α is sufficiently small, the reduction in P is guaranteed. Convergence to the desired solution is achieved when the inequality $P \le \epsilon$ is met, where ϵ is a small, preselected number.

The variations per unit stepsize $\Delta x/\alpha = A$ are governed by a system of n nonhomogeneous, linear differential equations subject to p separated initial conditions, q separated

[1] This research was supported by the National Science Foundation, Grant No. GP-18522. The authors are indebted to Mr. A. V. Levy for analytical and computational assistance.

[2] Professor of Astronautics and Director of the Aero-Astronautics Group, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

[3] Graduate Student in Aero-Astronautics, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

NGR-44-006-089

final conditions, and r mixed boundary conditions. This system is solved employing the method of particular solutions : $q + r + 1$ independent solutions are combined linearly, and the coefficients of the combination are determined so that the linear system is satisfied.

Several numerical examples are presented. They illustrate (i) the simplicity as well as the rapidity of convergence of the modified quasilinearization algorithm and (ii) the importance of stepsize control.

1. Introduction

In recent years, considerable attention has been devoted to the solution of the two-point boundary-value problem for nonhomogeneous, linear differential systems. Among the techniques available, we mention (a) the method of adjoint variables and (b) the method of complementary functions (Ref. 1). Methods (a) and (b) have one common characteristic: each requires the solution of two differential systems, namely, the original system plus the derived system; this derived system is the adjoint system in Case (a) and the homogeneous system in Case (b).

With particular regard to high-speed digitial computing, programming can be made simpler if one employs the original system only. This technique, a modification of (b), consists of combining linearly several particular solutions of the original, nonhomogeneous system. For this reason, it has been called the method of particular solutions (Ref. 2). It has the following advantages with respect to the previous techniques: (α) it makes use of only one differential system, namely, the original, nonhomogeneous system; (β) each particular solution satisfies the same prescribed initial conditions; and (γ) in a physical problem, each particular solution represents a physically possible trajectory, even though it satisfies only the initial conditions and not the final conditions.

While the method of particular solutions has been developed for linear systems, it can also be used to solve nonlinear systems. First, quasilinearization must be employed, and the nonlinear system must be replaced by one that is linear in the perturbation about a nominal function (see, for example, Refs. 3-6); to this linear system, the method of particular solutions can be applied to find the perturbation leading to a new nominal function; then, the procedure is employed iteratively (Ref. 7).

The main advantage of the ordinary quasilinearization algorithm is simplicity and rapidity of convergence if the nominal function is a fair approximation to the solution. There are cases, however, where ordinary quasilinearization diverges due to the excessive magnitude of the variations. This is why it is convenient to imbed the linearized system into a more general system by means of the scaling factor α, $0 \le \alpha \le 1$, applied to each forcing term. The resulting algorithm is called the modified quasilinearization algorithm.

At first glance, the above imbedding procedure seems arbitrary. However, a rigorous conceptual justification can be given through the consideration of the performance index P: this is the cumulative error in the differential equations and the boundary conditions. By computing the first variation of the functional P and requiring δP to be negative, one generates the modified quasilinearization algorithm. The main property of this algorithm is the descent property: if the stepsize α is sufficiently small, the reduction in P is guaranteed. In addition, the performance index P can also be employed as a convergence criterion: the algorithm is terminated when P becomes smaller than some preselected value.

2. Modified Quasilinearization

Consider a system described by the differential equation

$$\dot{x} - \varphi(x, t) = 0 \quad , \quad 0 \le t \le 1 \tag{1}$$

subject to the boundary conditions

$$f[x(0)] = 0 \quad , \quad g[x(1)] = 0 \quad , \quad h[x(0), x(1)] = 0 \tag{2}$$

Here, x and φ are n-vectors, f is a p-vector, g a q-vector, and h an r-vector, with $p + q + r = n$. The time t, a scalar, is the independent variable; without loss of generality, the prescribed initial time is $t = 0$ and the prescribed final time is $t = 1$. The dot denotes a derivative with respect to t.

It is assumed that (a) the first derivative of the function φ with respect to the vector x exists and is continuous and (b) the first derivatives of the functions f, g, h with respect to the vectors $x(0)$ and $x(1)$ exist and are continuous. It is also assumed that a solution of Eqs. (1)-(2) exists. The problem is to find the continuous vector function $x(t)$ which solves Eqs. (1)-(2).

2.1. Performance Index. In general, the system (1)-(2) is nonlinear, so that approximate methods must be employed. In this connection, consider the class of continuous functions $x(t)$ not necessarily satisfying Eqs. (1)-(2). For these functions, let the performance index P be defined as[4]

$$P = \int_0^1 (\dot{x} - \varphi)^T (\dot{x} - \varphi) dt + (f^T f + g^T g + h^T h) \tag{3}$$

The scalar functional P measures the cumulative error in the differential equation (1) and the boundary conditions (2); therefore, $P = 0$ for any $x(t)$ satisfying Eqs. (1)-(2) and

[4] The superscript T denotes the transpose of a matrix.

$P > 0$ otherwise. When approximate methods are used, they must ultimately lead to a state $x(t)$ such that

$$P \leq \epsilon \tag{4}$$

where ϵ is a small, preselected number.

2.2. Modified Quasilinearization. Here, we present a modification of the quasilinearization algorithm which has a descent property in the performance index P. Consider a nominal function $x(t)$ and a varied function $\tilde{x}(t)$ such that

$$\tilde{x}(t) = x(t) + \Delta x(t) \tag{5}$$

where $\Delta x(t)$ denotes the perturbation of x at a constant station t. The passage from the nominal function to the varied function causes the performance index P to change. To first order, we see that

$$\delta P = 2\int_0^1 (\dot{x} - \varphi)^T \delta(\dot{x} - \varphi)dt + 2(f^T \delta f + g^T \delta g + h^T \delta h) \tag{6}$$

where the symbol $\delta(\ldots)$ denotes the first variation.

Next, consider the system of variations defined by

$$\delta(\dot{x} - \varphi) = -\alpha(\dot{x} - \varphi) \tag{7}$$

and

$$\delta f = -\alpha f \ , \quad \delta g = -\alpha g \ , \quad \delta h = -\alpha h \tag{8}$$

where α is a scaling factor (or stepsize) in the range

$$0 \leq \alpha \leq 1 \tag{9}$$

Consequently, the first variation of the performance index P becomes

$$\delta P = -2\alpha \int_0^1 (\dot{x} - \varphi)^T (\dot{x} - \varphi) dt - 2\alpha (f^T f + g^T g + h^T h) \tag{10}$$

and, in the light of the definition (3), is equivalent to

$$\delta P = -2\alpha P \tag{11}$$

Note that, for any nominal curve x(t) not satisfying Eqs. (1)-(2),

$$P > 0 \tag{12}$$

Therefore, for α positive, one has

$$\delta P < 0 \tag{13}$$

This is the basic descent property of the algorithm defined by Eqs. (7)-(8); it guarantees that, if α is sufficiently small,

$$\tilde{P} < P \tag{14}$$

2.3. <u>System of Variations.</u> To first order, the changes of the functions appearing in Eqs. (7) are related to the change $\Delta x(t)$ as follows:[5]

$$\delta(\dot{x} - \varphi) = \Delta\dot{x} - \varphi_x^T \Delta x \quad , \quad 0 \le t \le 1 \tag{15}$$

---

[5] The matrix φ_x appearing in Eq. (15) is defined so that its ith column is the gradient of the ith scalar component of φ with respect to the vector x. Analogous definitions hold for the matrices appearing in Eqs. (16).

and

$$\delta f = f^T_{x(0)} \Delta x(0)$$

$$\delta g = g^T_{x(1)} \Delta x(1) \tag{16}$$

$$\delta h = h^T_{x(0)} \Delta x(0) + h^T_{x(1)} \Delta x(1)$$

where the matrix φ_x is n x n, the matrix $f_{x(0)}$ is n x p, the matrix $g_{x(1)}$ is n x q, and the matrices $h_{x(0)}$ and $h_{x(1)}$ are n x r. Consequently, Eqs. (7)-(8) can be rewritten as

$$\Delta\dot{x} - \varphi_x^T \Delta x + \alpha(\dot{x} - \varphi) = 0 \quad , \qquad 0 \le \alpha \le 1 \tag{17}$$

and

$$f^T_{x(0)} \Delta x(0) + \alpha f = 0$$

$$g^T_{x(1)} \Delta x(1) + \alpha g = 0 \tag{18}$$

$$h^T_{x(0)} \Delta x(0) + h^T_{x(1)} \Delta x(1) + \alpha h = 0$$

For a given value of α, Eq. (17) is equivalent to n scalar differential equations and Eqs. (18) are equivalent to p + q + r = n scalar boundary conditions. These equations and boundary conditions are linear and nonhomogeneous in the n components of the vector $\Delta x(t)$. The resulting algorithm is called modified quasilinearization algorithm.

For $\alpha = 1$, Eqs. (17)-(18) become identical with those of ordinary quasilinearization (Refs. 3-6), that is, the equations obtained by linearizing Eqs. (1)-(2) about the nominal function x(t). While modified quasilinearization exhibits the descent property (13)-(14), this is not necessarily the case with ordinary quasilinearization. This means that, if Eqs. (17)-(18) are employed with $\alpha = 1$, the performance index P may actually increase when passing from the nominal function x(t) to the varied function $\tilde{x}(t)$.

2.4. Coordinate Transformation. To simplify the problem, we introduce the auxiliary variable

$$A = \Delta x/\alpha \tag{19}$$

and rewrite Eqs. (17)-(18) in the form

$$\dot{A} - \varphi_x^T A + \dot{x} - \varphi = 0 \quad , \quad 0 \le t \le 1 \tag{20}$$

and

$$\begin{aligned} f_{x(0)}^T A(0) + f &= 0 \\ g_{x(1)}^T A(1) + g &= 0 \\ h_{x(0)}^T A(0) + h_{x(1)}^T A(1) + h &= 0 \end{aligned} \tag{21}$$

The differential system (20)-(21) is linear and nonhomogeneous in the function $A(t)$ and can be solved without assigning a value to the stepsize α. With $A(t)$ known (see Section 2.5) and the stepsize α specified (see Section 2.6), the correction $\Delta x(t)$ is obtained from (19), and the varied function $\tilde{x}(t)$ is computed from (5).

2.5. Integration Technique. Assuming that $p \ge q$, we integrate the previous differential system $q + r + 1$ times using a forward integration scheme in combination with the method of particular solutions (Ref. 2). In each integration, we specify the initial conditions[6]

$$A_i^j(0) = \delta_{ij} \quad , \quad i = 1, 2, \ldots, q + r + 1 \quad , \quad j = 1, 2, \ldots, q + r \tag{22}$$

[6] The subscript i denotes a particular integration. The superscript j denotes a particular component of the vector A.

where the Kronecker delta δ_{ij} is such that

$$\begin{aligned} \delta_{ij} &= 1 \quad , \quad i = j \\ \delta_{ij} &= 0 \quad , \quad i \neq j \end{aligned} \tag{23}$$

Then, we compute the missing initial conditions

$$A_i^j(0) \quad , \quad i = 1, 2, \ldots, q + r + 1 \quad , \quad j = q+r+1,\ q+r+2, \ldots, n \tag{24}$$

by solving Eq. (21-1). After performing the forward integrations, we obtain the functions

$$A_i = A_i(t) \quad , \quad i = 1, 2, \ldots, q + r + 1 \tag{25}$$

each of which satisfies (20) and (21-1) but not necessarily (21-2) and (21-3).

Next, we introduce the $q + r + 1$ undetermined, scalar constants k_i and form the linear combination

$$A(t) = \sum_{i=1}^{q+r+1} k_i A_i(t) \tag{26}$$

Then, we inquire whether, by an appropriate choice of the constants k_i, this linear combination can satisfy Eqs. (20)-(21). By simple substitution, it can be verified that the linear combination (26) satisfies the differential equation (20) and the separated initial condition (21-1) providing

$$\sum_{i=1}^{q+r+1} k_i = 1 \tag{27}$$

Finally, the function (26) satisfies the separated final condition (21-2) and the mixed boundary condition (21-3) providing

$$\sum_{i=1}^{q+r+1} k_i [g_{x(1)}^T A_i(1)] + g = 0$$

$$\sum_{i=1}^{q+r+1} k_i [h_{x(0)}^T A_i(0) + h_{x(1)}^T A_i(1)] + h = 0 \tag{28}$$

The linear system (27)-(28) is equivalent to $q + r + 1$ scalar equations, in which the unknowns are the $q + r + 1$ scalar constants k_i. After the constants k_i are known, the function $A(t)$ is computed with (26). In this way, the two-point boundary-value problem is solved.

2.6. Determination of the Stepsize. After combining Eqs. (5) and (19), we obtain the relation

$$\tilde{x}(t) = x(t) + \alpha A(t) \tag{29}$$

Since the function $x(t)$ is given and the function $A(t)$ is known by solving the linearized, two-point boundary-value problem, Eq. (29) yields a one-parameter family of solutions, the parameter being the stepsize α. For this one-parameter family, the performance index P becomes a function of the form

$$P = P(\alpha) \tag{30}$$

At $\alpha = 0$, the slope of this function is negative and is given by

$$P_\alpha(0) = -2P(0) \tag{31}$$

The function (30) exhibits a relative minimum with respect to α, that is, a point where

$$P_\alpha(\alpha) = 0 \tag{32}$$

This point can be determined by means of a one-dimensional search (for example, using quadratic interpolation, cubic interpolation, or quasilinearization). Ideally, this procedure should be used iteratively until the modulus of the slope satisfies the following inequality:

$$|P_\alpha(\alpha)| \le \theta \tag{33}$$

where θ is a small, preselected number.

Since the rigorous determination of α takes time on a computer, one might renounce solving Eq. (32) with a particular degree of precision and determine the stepsize in a noniterative fashion. To this effect, we first assign the value

$$\alpha = 1 \tag{34}$$

to the stepsize; this corresponds to full quasilinearization of Eqs. (1)-(2) and is the value which would solve Eq. (32) exactly, should Eqs. (1)-(2) be linear. Of course, the stepsize is acceptable only if

$$P(\alpha) < P(0) \tag{35}$$

Otherwise, the previous value of α must be replaced by some smaller value in the range (9) (for example, using a bisection process) until Ineq. (35) is met. This is guaranteed by the descent property (13)-(14).

2.7. <u>Summary of the Algorithm.</u> In the light of the previous discussion, we summarize the <u>modified quasilinearization</u> as follows:

(a) Assume a nominal function $x(t)$.

(b) Along the interval of integration, compute the vector $\dot{x} - \varphi$ and the matrix φ_x. On the boundary, compute the vectors f, g, h and the matrices $f_{x(0)}$, $g_{x(1)}$, $h_{x(0)}$, $h_{x(1)}$.

(c) Solve the linearized two-point boundary-value problem (20)-(21) using the forward integration scheme of Section 2.5.

(d) Consider the one-parameter family of the solutions (29) and perform a one-dimensional search on the function (30); specifically, perform a bisection process on α (starting from $\alpha = 1$), and continue the process until Ineq. (35) is satisfied.

(e) Once the stepsize α is known, compute the varied function $\tilde{x}(t)$ from (29).

(f) With the varied function known, the iteration is completed. The varied function $\tilde{x}(t)$ becomes the nominal function $x(t)$ for the next iteration. That is, return to (a) and iterate the algorithm.

(g) The algorithm is terminated when the stopping condition (4) is satisfied.

3. Numerical Examples[7]

In order to illustrate the theory, several numerical examples were developed using a Burroughs B-5500 computer and double-precision arithmetic. The algorithm was programmed in FORTRAN IV. The interval of integration was divided into 100 steps for the first five examples, 200 steps for the sixth example, and 500 steps for the seventh example. The differential system (20)-(21) was integrated using Hamming's modified predictor-corrector method with a special Runge-Kutta procedure to start the integration routine (Ref. 8). The definite integral P was computed using Simpson's rule.

Convergence was defined as follows:

$$P \leq 10^{-16} \tag{36}$$

and the number of iterations at convergence N_* was recorded. Conversely, nonconvergence was defined by means of the inequalities

(a) $$N \geq 40 \tag{37}$$

or

(b) $$N_S \geq 10 \tag{38}$$

or

(c) $$M \geq 0.4 \times 10^{69} \tag{39}$$

Here, N is the iteration number, N_S is the number of bisections of the stepsize α (starting from $\alpha = 1$) required to satisfy Ineq. (35), and M is the modulus of any of the quantities

---

[7] For simplicity, the symbols employed in this section denote scalar quantities.

employed in the algorithm. Satisfaction of Ineq. (37) indicates divergence or extreme slowness of convergence; in turn, satisfaction of Ineq. (38) indicates extreme smallness of the displacement Δx; finally, satisfaction of Ineq. (39) indicates exponential overflow for the Burroughs B-5500 computer: the computer program is automatically stopped.

Example 3.1. Consider the differential equations

$$\dot{x} = 3y \quad , \quad \dot{y} = -3 \sin x \tag{40}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad x(1) = 3 \tag{41}$$

In this problem, $n = 2$, $p = 1$, $q = 1$, $r = 0$. Since $q + r + 1 = 2$, two particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 3t \quad , \quad y(t) = 0 \tag{42}$$

which are consistent with the boundary conditions (41) but are not consistent with the differential equations (40). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 4$ iterations. The numerical results are presented in Tables 1-2, where N denotes the iteration number[8].

[8] In Tables 1-2 as well as subsequent tables, all data are truncated rather than rounded-off.

Table 1. Stepsize and performance index (Example 3.1).

N	α	P
0	–	0.1×10^{2}
1	1	0.7×10^{0}
2	1	0.1×10^{-2}
3	1	0.1×10^{-7}
4	1	0.4×10^{-18}

Table 2. Converged solution (Example 3.1, N = 4).

t	x	y
0.0	0.0000×10^{0}	0.2011×10^{1}
0.1	0.5944×10^{0}	0.1923×10^{1}
0.2	0.1140×10^{1}	0.1696×10^{1}
0.3	0.1606×10^{1}	0.1404×10^{1}
0.4	0.1983×10^{1}	0.1114×10^{1}
0.5	0.2278×10^{1}	0.8624×10^{0}
0.6	0.2505×10^{1}	0.6596×10^{0}
0.7	0.2679×10^{1}	0.5042×10^{0}
0.8	0.2812×10^{1}	0.3895×10^{0}
0.9	0.2916×10^{1}	0.3080×10^{0}
1.0	0.3000×10^{1}	0.2537×10^{0}

Example 3.2. Consider the differential equations[9]

$$\dot{x} = 2\pi y \quad , \quad \dot{y} = -2\pi[6x + x^2 + \cos(2\pi t)] \tag{43}$$

subject to the boundary conditions

$$x(0) = x(1) \quad , \quad y(0) = y(1) \tag{44}$$

In this problem, $n = 2$, $p = 0$, $q = 0$, $r = 2$. Since $q + r + 1 = 3$, three particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 0 \quad , \quad y(t) = 0 \tag{45}$$

which are consistent with the boundary conditions (44), but are not consistent with the differential equations (43). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 3$ iterations. The numerical results are presented in Tables 3-4, where N denotes the iteration number.

---

[9] This example has been considered in Ref. 9.

Table 3. Stepsize and performance index (Example 3.2).

N	α	P
0	–	0.1×10^{2}
1	1	0.2×10^{-1}
2	1	0.2×10^{-6}
3	1	0.3×10^{-16}

Table 4. Converged solution (Example 3.2, N = 4).

t	x	y
0.0	-0.2134×10^{0}	-0.7650×10^{-7}
0.1	-0.1690×10^{0}	0.1351×10^{0}
0.2	-0.5780×10^{-1}	0.2037×10^{0}
0.3	0.6730×10^{-1}	0.1802×10^{0}
0.4	0.1560×10^{0}	0.9693×10^{-1}
0.5	0.1865×10^{0}	-0.1049×10^{-7}
0.6	0.1560×10^{0}	-0.9693×10^{-1}
0.7	0.6730×10^{-1}	-0.1802×10^{0}
0.8	-0.5780×10^{-1}	-0.2037×10^{0}
0.9	-0.1690×10^{0}	-0.1351×10^{0}
1.0	-0.2134×10^{0}	-0.7650×10^{-7}

Example 3.3. Consider the differential equations

$$\dot{x} = \frac{1}{2}x^2y \quad , \quad \dot{y} = -\frac{1}{2}xy^2 \tag{46}$$

subject to the boundary conditions

$$x(0) + x(1) - e - 1 = 0 \quad , \quad y(0) - x(1)y(1) = 0 \tag{47}$$

where $e = 2.71828$. In this problem, $n = 2$, $p = 0$, $q = 0$, $r = 2$. Since $q + r + 1 = 3$, three particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 2 \quad , \quad y(t) = 1 \tag{48}$$

which are not consistent with (46)-(47). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 5$ iterations. The numerical results are presented in Tables 5-6, where N denotes the iteration number.[10]

[10]The solution of problem (46)-(47) is not unique. Another solution is characterized by constant values of x and y, specifically, $x(t) = \frac{1}{2}(e + 1)$, $y(t) = 0$.

Table 5. Stepsize and performance index (Example 3.3).

N	α	P
0	–	0.6×10^{1}
1	1	0.8×10^{0}
2	1	0.1×10^{-1}
3	1	0.1×10^{-6}
4	1	0.8×10^{-15}
5	1	0.5×10^{-32}

Table 6. Converged solution (Example 3.3, N = 5).

t	x	y
0.0	0.1000×10^{1}	0.2000×10^{1}
0.1	0.1105×10^{1}	0.1809×10^{1}
0.2	0.1221×10^{1}	0.1637×10^{1}
0.3	0.1349×10^{1}	0.1481×10^{1}
0.4	0.1491×10^{1}	0.1340×10^{1}
0.5	0.1648×10^{1}	0.1213×10^{1}
0.6	0.1822×10^{1}	0.1097×10^{1}
0.7	0.2013×10^{1}	0.9931×10^{0}
0.8	0.2225×10^{1}	0.8986×10^{0}
0.9	0.2459×10^{1}	0.8131×10^{0}
1.0	0.2718×10^{1}	0.7357×10^{0}

Example 3.4. Consider the differential equations

$$\dot{x} = y \quad , \quad \dot{y} = z \quad , \quad \dot{z} = -z^2 uw/6$$
$$\dot{u} = w \quad , \quad \dot{w} = -yw^3/2 \tag{49}$$

subject to the boundary conditions

$$x(0) = 1 \quad , \quad u(0) = 1 \quad , \quad w(0) = -1 \quad , \quad x(1) = 16 \quad , \quad u(1) = 1/2 \tag{50}$$

In this problem, $n = 5$, $p = 3$, $q = 2$, $r = 0$. Since $q + r + 1 = 3$, three particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 1 + 15t \quad , \quad y(t) = 0 \quad , \quad z(t) = 0$$
$$u(t) = 1 - t/2 \quad , \quad w(t) = -1 \tag{51}$$

which are consistent with the boundary conditions (50) but are not consistent with the differential equations (49). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 11$ iterations. The numerical results are presented in Tables 7-8, where N denotes the iteration number.

Table 7. Stepsize and performance index (Example 3.4).

N	α	P
0	–	0.2×10^{3}
1	1/16	0.2×10^{3}
2	1/8	0.1×10^{3}
3	1/4	0.1×10^{3}
4	1/2	0.4×10^{2}
5	1/2	0.2×10^{2}
6	1	0.1×10^{1}
7	1/2	0.3×10^{0}
8	1/2	0.1×10^{0}
9	1	0.2×10^{-1}
10	1	0.2×10^{-8}
11	1	0.1×10^{-21}

Table 8. Converged solution (Example 3.4, N = 11).

t	x	y	z	u	w
0.0	0.1000×10^1	0.4000×10^1	0.1200×10^2	0.1000×10^1	-0.1000×10^1
0.1	0.1464×10^1	0.5324×10^1	0.1452×10^2	0.9090×10^0	-0.8264×10^0
0.2	0.2073×10^1	0.6912×10^1	0.1728×10^2	0.8333×10^0	-0.6944×10^0
0.3	0.2856×10^1	0.8788×10^1	0.2028×10^2	0.7692×10^0	-0.5917×10^0
0.4	0.3841×10^1	0.1097×10^2	0.2352×10^2	0.7142×10^0	-0.5102×10^0
0.5	0.5062×10^1	0.1350×10^2	0.2700×10^2	0.6666×10^0	-0.4444×10^0
0.6	0.6553×10^1	0.1638×10^2	0.3072×10^2	0.6250×10^0	-0.3906×10^0
0.7	0.8352×10^1	0.1965×10^2	0.3468×10^2	0.5882×10^0	-0.3460×10^0
0.8	0.1049×10^2	0.2332×10^2	0.3888×10^2	0.5555×10^0	-0.3086×10^0
0.9	0.1303×10^2	0.2743×10^2	0.4332×10^2	0.5263×10^0	-0.2770×10^0
1.0	0.1600×10^2	0.3200×10^2	0.4800×10^2	0.5000×10^0	-0.2500×10^0

Example 3.5. Consider the differential equations

$$\dot{x} = 10y \;, \quad \dot{y} = 10z \;, \quad \dot{z} = -5xz \tag{52}$$

subject to the boundary conditions

$$x(0) = 0 \;, \quad y(0) = 0 \;, \quad y(1) = 1 \tag{53}$$

In this problem, $n = 3$, $p = 2$, $q = 1$, $r = 0$. Since $q + r + 1 = 2$, two particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 0 \;, \quad y(t) = t \;, \quad z(t) = 0 \tag{54}$$

which are consistent with the boundary conditions (53) but are not consistent with the differential equations (52). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 6$ iterations. The numerical results are presented in Tables 9-10, where N denotes the iteration number.

Table 9. Stepsize and performance index (Example 3. 5).

N	α	P
0	–	0.3×10^{2}
1	1	0.1×10^{1}
2	1/8	0.2×10^{0}
3	1	0.3×10^{-1}
4	1	0.5×10^{-4}
5	1	0.2×10^{-9}
6	1	0.2×10^{-20}

Table 10. Converged solution (Example 3.5, N = 6).

t	x	y	z
0.0	0.0000×10^{0}	0.0000×10^{0}	0.3320×10^{0}
0.1	0.1655×10^{0}	0.3297×10^{0}	0.3230×10^{0}
0.2	0.6500×10^{0}	0.6297×10^{0}	0.2667×10^{0}
0.3	0.1396×10^{1}	0.8460×10^{0}	0.1613×10^{0}
0.4	0.2305×10^{1}	0.9555×10^{0}	0.6423×10^{-1}
0.5	0.3283×10^{1}	0.9915×10^{0}	0.1590×10^{-1}
0.6	0.4279×10^{1}	0.9989×10^{0}	0.2402×10^{-2}
0.7	0.5279×10^{1}	0.9999×10^{0}	0.2201×10^{-3}
0.8	0.6279×10^{1}	0.9999×10^{0}	0.1224×10^{-4}
0.9	0.7279×10^{1}	0.9999×10^{0}	0.4130×10^{-6}
1.0	0.8279×10^{1}	0.1000×10^{1}	0.8413×10^{-8}

Example 3.6. Consider the differential equations[11]

$$\dot{x} = 13y, \ , \quad \dot{y} = 13z \ , \quad \dot{z} = -20.15xz + 1.3y^2 - 13u^2 + 2.6y + 13$$

$$\dot{u} = 13w \ , \quad \dot{w} = -20.15xw + 14.3yu + 2.6u - 2.6 \tag{55}$$

subject to the boundary conditions

$$x(0) = 0 \ , \quad y(0) = 0 \ , \quad u(0) = 0 \ , \quad y(1) = 0 \ , \quad u(1) = 1 \tag{56}$$

In this problem, $n = 5$, $p = 3$, $q = 2$, $r = 0$. Since $q + r + 1 = 3$, three particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = 0 \ , \quad y(t) = 0 \ , \quad z(t) = 0 \ , \quad u(t) = t \ , \quad w(t) = 0 \tag{57}$$

which are consistent with the boundary conditions (56) but are not consistent with the differential equations (50). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 6$ iterations. The numerical results are presented in Tables 11-12, where N denotes the iteration number.

[11] This example, which involes unstable differential equations, was considered in Ref. 10.

Table 11. Stepsize and performance index (Example 3.6).

N	α	P
0	–	0.9×10^{2}
1	1/2	0.3×10^{2}
2	1/2	0.8×10^{1}
3	1	0.4×10^{-1}
4	1	0.6×10^{-4}
5	1	0.5×10^{-10}
6	1	0.3×10^{-22}

Table 12. Converged solution (Example 3.6, N = 6).

t	x	y	z	u	w
0.0	0.0000×10^{0}	0.0000×10^{0}	-0.9663×10^{0}	0.0000×10^{0}	0.6529×10^{0}
0.1	-0.5028×10^{0}	-0.5802×10^{0}	-0.7188×10^{-1}	0.6971×10^{0}	0.4220×10^{0}
0.2	-0.1215×10^{1}	-0.4603×10^{0}	0.1945×10^{0}	0.1100×10^{1}	0.2036×10^{0}
0.3	-0.1631×10^{1}	-0.1744×10^{0}	0.2210×10^{0}	0.1247×10^{1}	0.3249×10^{-1}
0.4	-0.1688×10^{1}	0.7033×10^{-1}	0.1443×10^{0}	0.1213×10^{1}	-0.7189×10^{-1}
0.5	-0.1506×10^{1}	0.1844×10^{0}	0.3000×10^{-1}	0.1093×10^{1}	-0.1002×10^{0}
0.6	-0.1270×10^{1}	0.1602×10^{0}	-0.5755×10^{-1}	0.9815×10^{0}	-0.6490×10^{-1}
0.7	-0.1120×10^{1}	0.6614×10^{-1}	-0.7534×10^{-1}	0.9334×10^{0}	-0.1024×10^{-1}
0.8	-0.1091×10^{1}	-0.1365×10^{-1}	-0.4303×10^{-1}	0.9447×10^{0}	0.2223×10^{-1}
0.9	-0.1133×10^{1}	-0.4258×10^{-1}	-0.1453×10^{-2}	0.9774×10^{0}	0.2352×10^{-1}
1.0	-0.1173×10^{1}	-0.1508×10^{-20}	0.9405×10^{-1}	0.1000×10^{1}	0.1765×10^{-1}

Example 3.7. Consider the differential equations[12]

$$\dot{x} = u \quad , \quad \dot{y} = w$$
$$\dot{u} = 2w + f_x \quad , \quad \dot{w} = -2u + f_y \tag{58}$$

where

$$f = (x^2 + y^2)/2 + (1 - \mu)/r + \mu/\rho + \mu(1 - \mu)/2$$
$$r = \sqrt{[(x - \mu)^2 + y^2]} \quad , \quad \rho = \sqrt{[(x + 1 - \mu)^2 + y^2]} \tag{59}$$

and where $\mu = 0.012$. These equations are subject to the boundary conditions

$$x(0) = -0.2 \quad , \quad y(0) = -0.1 \quad , \quad x(1) = -1.2 \quad , \quad y(1) = 0 \tag{60}$$

In this problem, $n = 4$, $p = 2$, $q = 2$, $r = 0$. Since $q + r + 1 = 3$, three particular solutions are needed per iteration.

Assume the nominal functions

$$x(t) = -0.2 - t \quad , \quad y(t) = -0.1 + 0.1t$$
$$u(t) = -1 \quad , \quad w(t) = 0.1 \tag{61}$$

which are consistent with the boundary conditions (60) but are not consistent with the differential equations (58). Starting with these nominal functions, we employ the algorithm of Section 2. Convergence to the solution is achieved in $N_* = 7$ iterations. The numerical results are presented in Tables 13-14, where N denotes the iteration number.

[12] This example refers to the restricted three-body problem (Ref. 11).

Table 13. Stepsize and performance index (Example 3.7).

N	α	P
0	–	0.6×10^{2}
1	1/4	0.5×10^{2}
2	1/64	0.5×10^{2}
3	1/8	0.4×10^{2}
4	1	0.3×10^{0}
5	1	0.6×10^{-5}
6	1	0.5×10^{-13}
7	1	0.2×10^{-29}

Table 14. Converged solution (Example 3.7, N = 7).

t	x	y	u	w
0.0	-0.2000×10^{0}	-0.1000×10^{0}	-0.1847×10^{1}	-0.1789×10^{1}
0.1	-0.3517×10^{0}	-0.2367×10^{0}	-0.1339×10^{1}	-0.1039×10^{1}
0.2	-0.4788×10^{0}	-0.3175×10^{0}	-0.1222×10^{1}	-0.6008×10^{0}
0.3	-0.5980×10^{0}	-0.3604×10^{0}	-0.1164×10^{1}	-0.2688×10^{0}
0.4	-0.7118×10^{0}	-0.3730×10^{0}	-0.1109×10^{1}	0.8213×10^{-2}
0.5	-0.8195×10^{0}	-0.3599×10^{0}	-0.1042×10^{1}	0.2497×10^{0}
0.6	-0.9196×10^{0}	-0.3240×10^{0}	-0.9576×10^{0}	0.4645×10^{0}
0.7	-0.1010×10^{1}	-0.2677×10^{0}	-0.8520×10^{0}	0.6572×10^{0}
0.8	-0.1089×10^{1}	-0.1932×10^{0}	-0.7207×10^{0}	0.8295×10^{0}
0.9	-0.1153×10^{1}	-0.1027×10^{0}	-0.5577×10^{0}	0.9739×10^{0}
1.0	-0.1200×10^{1}	0.0000×10^{0}	-0.3719×10^{0}	0.1073×10^{1}

4. Remarks

The following remarks are pertinent to the previous theoretical development.

Remark 4.1. If the stepsize is set at the constant value $\alpha = 1$, the modified quasilinearization algorithm of Section 2 reduces to the ordinary quasilinearization algorithm. While modified quasilinearization exhibits the descent property (13)-(14), this is not necessarily the case with ordinary quasilinearization. Therefore, in ordinary quasilinearization, the performance index P may actually increase when passing from the nominal function x(t) to the varied function $\tilde{x}(t)$.

With reference to the examples of Section 3, computer runs were made employing both modified quasilinearization and ordinary quasilinearization: in Table 15, the number of iterations at convergence N_* is indicated and, as the table shows, the experimental evidence is in favor of modified quasilinearization. It is emphasized that the above conclusion was obtained through particular examples and that, consequently, the subject requires further investigation.

Remark 4.2. The fundamental property of the modified quasilinearization algorithm is the descent property (13)-(14). This local property guarantees the decrease of the performance index P when passing from the nominal function x(t) to the varied function $\tilde{x}(t)$. However, it does not guaranteee convergence; that is, it does not guarantee that $P \to 0$ as $N \to \infty$. This is due to the fact that convergence depends on the analytical nature of the functions φ, f, g, h and on the nominal function x(t) chosen in order to start the algorithm.

Table 15. Number of iterations for convergence.

	N_*	
	$\alpha \leq 1$	$\alpha = 1$
Example 3.1	4	4
Example 3.2	3	3
Example 3.3	5	5
Example 3.4	11	Nonconvergence (c)
Example 3.5	6	8
Example 3.6	6	Nonconvergence (c)
Example 3.7	7	5

5. Discussion and Conclusions

In this paper, a general method for solving nonlinear, two-point boundary-value problems is presented; it is assumed that the differential system has order n and is subject to p separated initial conditions, q separated final conditions, and r mixed boundary conditions, with $p + q + r = n$. The method is based on the consideration of the performance index P, the cumulative error in the differential equations and the boundary conditions.

A modified quasilinearization algorithm is generated by requiring the first variation of the performance index δP to be negative. The algorithm has the form $\tilde{x}(t) = x(t) + \alpha A(t)$. Here, α, $0 \le \alpha \le 1$, is the stepsize and the function $A(t)$ is obtained by solving a system of n differential equations subject to p separated initial conditions, q separated final conditions, and r mixed boundary conditions. In general, the differential equations and the boundary conditions are linear and nonhomogeneous. This system is solved employing the method of particular solutions: $q + r + 1$ independent solutions are combined linearly, and the coefficients of the combination are determined so that the linear system is satisfied.

The main property of the modified quasilinearization algorithm is the descent property: if the stepsize α is sufficiently small, the reduction in P is guaranteed. Not only is P employed as a guide during progression of the algorithm, but also as a convergence criterion: the algorithm is terminated when the performance index P becomes smaller than some preselected value.

Several numerical examples are presented; they illustrate (i) the simplicity as well as the rapidity of convergence of the algorithm and (ii) the importance of stepsize control.

References

1. GOODMAN, T.R., and LANCE, C.N., The Numerical Integration of Two-Point Boundary-Value Problems, Mathematical Tables and Other Aids to Computation, Vol. 10, No. 54, 1956.

2. MIELE, A., Method of Particular Solutions for Linear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968.

3. BELLMAN, R.E., and KALABA, R.E., Quasilinearization and Nonlinear Boundary-Value Problems, The RAND Corporation, Report No. R-438-PR, 1965.

4. LEE, E.S., Quasilinearization and Invariant Imbedding, Academic Press, New York, 1968.

5. KELLER, H.B., Numerical Methods for Two-Point Boundary-Value Problems, Blaisdell Publishing Company, Waltham, Massachusetts, 1968.

6. BAILEY, P.B., SHAMPINE, L.F., and WALTMAN, P.E., Nonlinear, Two-Point Boundary-Value Problems, Academic Press, New York 1968.

7. MIELE, A., and IYER, R.R., General Technique for Solving Nonlinear, Two-Point Boundary-Value Problems Via the Method of Particular Solutions, Journal of Optimization Theory and Applications, Vol. 5, No. 5, 1970.

8. RALSTON, A., Numerical Integration Methods for the Solution of Ordinary Differential Equations, Mathematical Methods for Digital Computers, Vol. 1, Edited by A. Ralston and H.S. Wilf, John Wiley and Sons, New York, 1960.

9. FALB, P.L., and DE JONG, J.L., Some Successive Approximation Methods in Control and Oscillation Theory, Academic Press, New York, 1969.

10. ROBERTS, S.M., SHIPMAN, J.S., and ROTH, C.D., Continuation in Quasilinearization, Journal of Optimization Theory and Applications, Vol. 2, No. 3, 1968.

11. SZEBEHELY, V., Theory of Orbits, Academic Press, New York, 1967.